# Aprendizado Multirrótulo para a Classificação Automática de Textos

**Luiz Henrique A. dos Santos, Rafael G. Rossi**

[1]Universidade Federal do Mato Grosso do Sul (UFMS)
UNID. II: Av. Ranulpho Marques Leal, 3484 – CEP 79620-080 – Cx Postal nº 210
Mato Grosso do Sul – MS – Brazil

luizhenriquesantos.lh@gmail.com, rafael.g.rossi@ufms.br

***Resumo.*** *A mineração de dados têm crescido no meio acadêmico e empresarial devido a popularização da internet e à produção massiva de dados. Grande parte destes dados estão em formato textual, como e-mails, postagens em redes sociais, notícias ou relatórios. Com a necessidade de gerenciar ou extrair conhecimento de grandes quantidades de textos, foi desenvolvida a classificação automática de textos, a qual é viabilizada por meio do uso de algoritmos de aprendizado de máquina. Porém, grande parte dos algoritmos aplicados em pesquisa ou situações práticas é de aprendizado multi-classe, isto é, considera que os textos pertencem a uma única dentre várias categorias. Entretanto, na prática, um texto pode pertencer a mais de uma dentre várias categorias. Por exemplo, um texto pode falar de esporte e economia. Para tal situação, deve-se fazer uso dos algoritmos de classificação mulitirrótulo. Duas abordagens de algoritmos de aprendizado multirrótulos podem ser utilizadas: (i) algoritmos de adaptação do problema; e (ii) algoritmos próprios para aprendizado multirrótulo. Os algoritmos de adaptação do problema consistem em transformar o aprendizado multirrótulo em aprendizado multi-classe, ou em vários problemas de aprendizado binário. Com isso, algoritmos de aprendizado multi-classe ou binário tradicionais podem ser utilizados. Já os algoritmos próprios são desenvolvidos para predizer automaticamente múltiplos rótulos para um documento. Porém, dada a grande variedade de opções para ambos os tipos de aprendizado multirrótulo e a ausência de estudos na literatura sobre os custo-benefícios das abordagens para a classificação automática de textos, o objetivo deste trabalho é realizar uma extensa avaliação experimental sobre aprendizado multirrótulo para a classificação automática de textos. Com os resultados obtidos pode-se concluir que técnicas de adaptação de problema, que transformam algoritmos multi-classe em multirrótulo, podem apresentar performance de classificação superior aos algoritmos próprios para aprendizado multirrótulo. Outro resultado interessante obtido é que o método de adaptação de problema Binary Relevance apresentou resultados competitivos em relação ao método de adaptação de problema Classifier Chain, sendo que o primeiro apresenta menor custo computacional. Além disso, vale ressaltar que dependendo do algoritmo de aprendizado indutivo utilizado, as abordagens de adaptação do problema podem ser mais rápidas que os algoritmos próprios.*

# 1. Introdução

Com o desenvolvimento e maior acesso às tecnologias de informação, o número de dados produzidos e disponibilizados tem crescido de forma exponencial. Segundo estudo realizado em 2014, de 2013 a 2020 o número de dados digitais aumentará de 4,4 trilhões de gigabytes para 44 trilhões de gigabytes [Turner et al. 2014]. Como a maioria do tempo na internet é gasto em redes sociais, grande parte desses dados estão em formato de texto, seja por meio de *tweets*, e *posts* em redes sociais. Além das redes sociais, dados textuais também são facilmente encontrados em e-mails corporativos e pessoais, notícias, relatórios e artigos.

Desperdiçar esses dados não é prudente seja por parte de meios acadêmicos ou por meios empresariais, uma vez que podem ser extraídas informações valiosas para ambos os meios. Por exemplo, é possível verificar a aceitação de produtos, a reputação de personalidades ou empresas, quantificar tipos de reclamações em empresas, redirecionar e-mails e organizar projetos [Weiss et al. 2015]. Porém, o grande volume de dados impossibilita a organização, gerenciamento e extração de conhecimento de forma manual.

Neste cenário, algoritmos de aprendizado de máquina têm sido utilizados para automatizar as tarefas mencionadas acima [Alpaydin 2009]. Os algoritmos de aprendizado de máquina são capazes de aprender a generalizar ou extrair padrões com base nos textos e seus respectivos rótulos (descritores de categorias). Apesar de viabilizar a classificação automática de textos, normalmente são aplicados algoritmos de aprendizado multi-classe tanto em pesquisas quanto em aplicações práticas [Manning et al. 2010, Sebastiani 2002, Dumais et al. 1998]. A característica desse tipo de aprendizado é que os classificadores gerados são capazes de atribuir apenas um rótulo a um texto não rotulado.

Entretanto, textos podem possuir mais de um rótulo ”por exemplo, podemos ter um texto de notícia que se refere a esporte e economia ao mesmo tempo”. Para atender tal cenário, pode-se fazer uso do aprendizado multirrótulo. Este tipo de aprendizado consiste em um conjunto de técnicas capazes de atribuir a um novo texto não rotulado, mais de um rótulo, a partir de um conjunto de treino anteriormente apresentado [Tsoumakas et al. 2009b].

O aprendizado mutirrótulo pode ser dividido em duas categorias de algoritmos: (i) algoritmos de adaptação do problema, e (ii) algoritmos próprios. Algoritmos de adaptação do problema consistem em transformar o aprendizado multirrótulo em aprendizado multiclasse, ou em vários problemas de aprendizado binário. Por exemplo, pode-se fazer com que os múltiplos rótulos formem um único rótulo, ou gerar uma base para cada rótulo, sendo que para cada base, um rótulo da coleção é considerado com classe positiva e os demais são considerados como negativos. Já os algoritmos próprios são capazes de induzir um modelo de classificação para inferir múltiplos rótulos sem a necessidade de modificar/adaptar rótulos do conjunto de treinamento.

Apesar dos benefícios do aprendizado multirrótulo, não há na literatura avaliações conclusivas considerando a diversidade de algoritmos multirrótulo e uma grande diversidade de bases textuais. Dado isso, o objetivo deste trabalho de conclusão de curso é realizar uma extensa avaliação experimental sobre aprendizado multirrótulo para a classificação automática de textos, e ao final, apresentar uma análise considerando a performance de classificação e o custo computacional.

O restante deste trabalho de conclusão de curso está dividido da seguinte forma. Na Seção 2 são apresentados os conceitos necessários para o entendimento deste trabalho. Na Seção 3 é apresentado o método de pesquisa e os detalhes de cada passo do método. Na Seção 4 são apresentados os resultados e discussões. Por fim, na Seção 5 são apresentadas as considerações finais e trabalhos futuros.

## 2. Conceitos

Nesta seção são apresentados os conceitos necessários para o entendimento deste trabalho. Primeiramente é apresentado como é representada uma coleção de texto para que os algoritmos de aprendizado de máquina possam interpretá-la. Posteriormente é apresentado o aprendizado multi-classe, o qual é tradicionalmente utilizado na classificação de texto e que pode ser também utilizado no aprendizado multirrótulo. Por fim, é apresentada uma seção referente ao aprendizado mutirrótulo, incluindo detalhes dos algoritmos de adaptação do problema e algoritmos próprios.

### 2.1. Representação estruturada das coleções de textos

Para utilizar algoritmos multirrótulo, é necessária uma estruturação da coleção de textos, de modo que os algoritmos sejam capazes de interpretar tais dados. Porém, antes de realizar a estruturação destes textos são necessários alguns passos de padronização e limpeza dos mesmos. Nesta etapa, é definido um padrão de caixa para o texto, para que as palavras não sejam diferenciadas por letras maiúsculas ou minúsculas, e além disso, são removidas *stopwords* ou seja, palavras que são consideradas desnecessárias para o entendimento do texto, para que estas, não influenciam no resultado final da classificação. Outro passo é a simplificação das palavras, que consistem em simplificar a palavra o máximo possível, sem que esta perca o significado (ex: radicalização). Isto se faz necessário para que as palavras não se diferenciam por tempo verbal, número ou gênero, de forma que palavras como alimentar, alimentos e alimentando correspondam a um único termo, por exemplo, *aliment* no caso da radicalização [Rossi et al. 2016].

Após o pré-processamento, o texto é transformado em um formato estruturado. A forma mais comum de se representar os textos é no modelo espaço vetorial, onde cada documento é representada por um vetor e cada dimensão corresponde a um atributo da coleção de textos. Quando os atributos correspondem a termos simples, a representação é denominada *bag-of-words*.

Na Tabela 1 é apresentada uma representação da representação bag-of-words, para uma coleção do $m$ documentos e $n$ termos. As células da matriz representam os pesos dos atributos nos documentos. Esse pesos geralmente são: binário, frequência do termo no documentos ($tf$), ou a $tf$ ponderada pelo inverso da frequência de ocorrência do termo na coleção (*tf-idf*) [Wallach 2006, Matsubara et al. 2003]. Além disso, no caso da classificação há uma coluna ou colunas adicionais para armazenar os rótulos dos documentos. Mais especificamente no aprendizado multirrótulo, há uma coluna para cara rótulo da coleção. Caso um documento pertença a um determinado rótulo, é definido o valor 1 na célula correspondente, e o valor 0 caso não pertença ao rótulo. Com isso, é possível definir mais de um rótulo para um documento apenas definindo o valor 1 nas células correspondentes.

**Tabela 1. Exemplo de *bag-of words* em formato de matriz documento-termo. Onde d representa os documentos, t os termos e r os rótulos**

| | $\mathbf{t_1}$ | $\mathbf{t_2}$ | $\mathbf{\ldots}$ | $\mathbf{t_n}$ | $\mathbf{r_1}$ | $\mathbf{r_2}$ | $\mathbf{r_3}$ | $\mathbf{\ldots}$ | $\mathbf{r_l}$ |
|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{d_1}$ | $w_{d_1,t_1}$ | $w_{d_1,t_2}$ | ... | $w_{d_1,t_n}$ | 1 | 0 | 0 | ... | 0 |
| $\mathbf{d_2}$ | $w_{d_2,t_1}$ | $w_{d_2,t_2}$ | ... | $w_{d_2,t_n}$ | 1 | 1 | 0 | ... | 1 |
| $\mathbf{d_3}$ | $w_{d_3,t_1}$ | $w_{d_3,t_2}$ | ... | $w_{d_3,t_n}$ | 0 | 1 | 0 | ... | 1 |
| $\mathbf{\vdots}$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $\mathbf{d_m}$ | $w_{d_m,t_1}$ | $w_{d_m,t_2}$ | ... | $w_{d_m,t_n}$ | 0 | 1 | 1 | ... | 0 |

## 2.2. Aprendizado multi-classe

Algoritmos de aprendizado multi-classe são algoritmos que conseguem atribuir apenas um rótulo para cada documento . Estes são importantes para este trabalho pois, uma vez que unidos com os algoritmos de transformação do problema, apresentados na Seção 2.3.1, podem ser utilizados para resolução de problemas multirrótulo. Na Figura 1 é apresentado um exemplo de aprendizado multi-classe

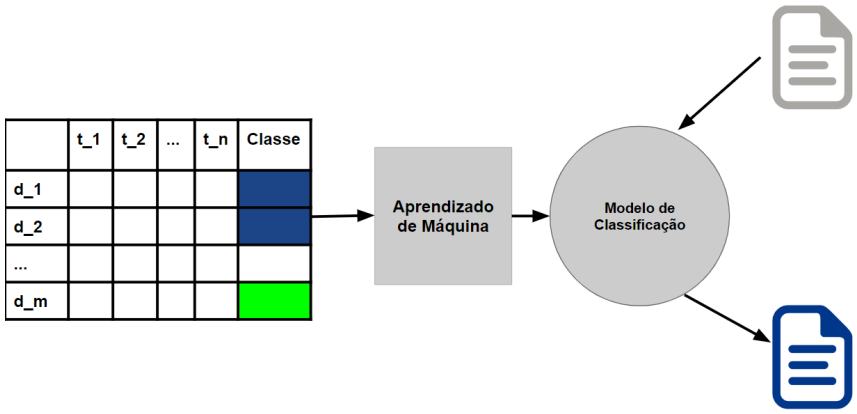



**Figura 1. Exemplo de aprendizado multi-classe. No qual $d$ representam os documentos, $t$ os termos, e as cores os rótulos dos documentos.**

## 2.3. Aprendizado multirrótulo

Na próxima subseção serão apresentados os detalhes dos algoritmos multirrótulo considerados para este trabalho divididos em duas categorias: (i) Algoritmos de transformação do Problema e (ii) Algoritmos Próprios [Zhang and Zhou 2013, Sorower 2010, Read et al. 2011].

### 2.3.1. Algoritmos de Transformação do Problema

Algoritmos de transformação do problema adaptam problemas multirrótulo para problemas multi-classe, o que permite que seja utilizada uma vasta gama de algoritmos já existentes na literatura de aprendizado multi-classe . Dentre os algoritmos de transformação do problema mais relevantes identificados na literatura, estão o *Binary Relevance* (BR), *Label PowerSet* (LP), *Ensemble of Pruned Sets* (EPS), *Classifier Chain* (CC) e *Multi-Label Stacking* (MLS), os quais estão presentes no *framework* produzido. Abaixo serão apresentados cada um destes algoritmos citados.

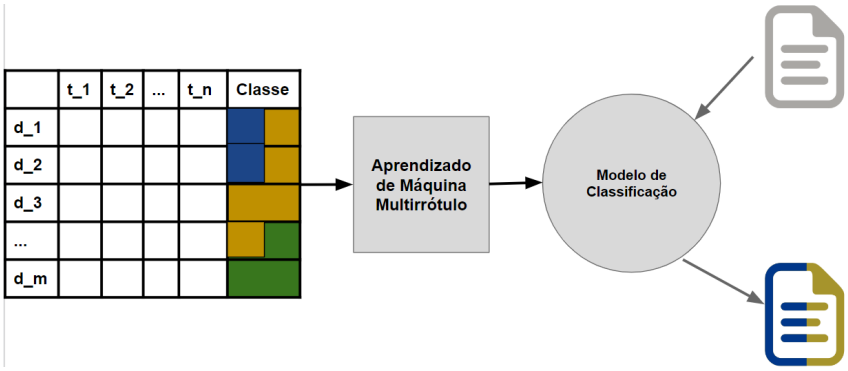



**Figura 2. Exemplo de algoritmo de aprendizado multirrótulo. No qual $d$ representam os documentos, $t$ os termos, e as cores os rótulos.**

**Binary Relevance**. A abordagem *BR* é uma das mais comuns na literatura. Esta, transforma o conjunto de dados em $|L|$ conjuntos e treina $|L|$ classificadores binários $H_l \rightarrow \{-1, 1\}$ para cada rótulo $l \in L$. Dado um novo documento, esse será rotulado com todos os rótulos dos classificadores que disparam a classe positiva [Cherman 2013]. Sua principal vantagem é sua baixa complexidade computacional em comparação com os outros métodos de transformação do problema. Porém, mesmo que o *Binary Relevance* seja mencionado em toda literatura e seja o mais comum e conhecido, em muitos casos é um método criticado com base no fato de que ele não modela diretamente as correlações que existem entre os rótulos nos dados de treinamento. Segundo alguns autores da literatura, a desconsideração dessas correlações entre os rótulos pode diminuir sua performance de classificação [Luaces et al. 2012]. Na Tabela 3 será ilustrado como ficaria a coluna correspondente à classe em para uma abordagem *Binary Relevance* considerando o rótulo $\mathbf{r_1}$ (apresentada na Tabela 1).

**Tabela 2. Exemplo de *Binary Relevance* considerando o rótulo $\mathbf{r_1}$ da Tabela 1**

| $\mathbf{r_1}$ |
|---|
| 1 |
| 1 |
| -1 |
| ... |
| -1 |

**Label Powerset**. A abordagem *LP* consiste em combinar conjuntos de rótulos e transformá-los em rótulos simples. Com isso, cada combinação de rótulos é considerada como sendo um único rótulo. Por exemplo, se um texto é rotulado como "Economia" e "Esporte", este possuirá o rótulo "Economia-Esporte" após a transformação do problema, ou seja, um conjunto multirrótulo com 10 rótulos pode ter até $2^{10} = 1024$ combinações de rótulos. Isso pode aumentar muito tempo de execução do aprendizado de máquina para alguns algoritmos, principalmente os algoritmos binários que são adaptados para o aprendizado multi-classe classificação.

A vantagem dessa abordagem é ser simples e permitir a aplicação de algoritmos

binários ou multi-classe existentes. Além disso pode solucionar um dos maiores problemas do $BR$, pois a classificação realizada no *BR* pode levar a uma situação em que um conjunto de rótulos é atribuído a uma instância, embora esses rótulos nunca coexistam juntos no conjunto de dados. Porém, algumas combinações de rótulos podem ser infrequentes no conjunto de dados, ocasionando assim um baixo número de exemplos para o aprendizado, o que pode afetar negativamente a performance de classificação. Além disso, a complexidade computacional dos algoritmos de aprendizado de máquina é diretamente proporcional ao número de rótulos. Na Tabela 3 é apresentado um exemplo de como ficariam os rótulos do exemplos apresentados na Tabela 1 considerando a abordagem *LP*.

**Tabela 3. Exemplo de *Label Powerset* para a Tabela 1**

| **LP** |
|---|
| $r_1$ |
| $r_1$-$r_2$-$r_n$ |
| $r_2$-$r_n$ |
| ... |
| $r_2$-$r_3$ |

**Ensembles of Pruned Sets**. A abordagem $EPS$ é uma extensão da abordagem $LP$ visando sanar algumas de suas limitações. Primeiramente, são considerados apenas combinações de rótulos frequentes para gerar rótulos únicos. Este fato diminui a complexidade computacional conforme mencionado na abordagem $LP$. Além disso, pode-se atribuir novas combinações de rótulos considerando valores da distribuição de probabilidades para os rótulos selecionando rótulos acima de um limiar, ou ainda gerando diferentes conjuntos de treinamentos para gerar diferentes classificadores e atribuir os rótulos de acordo com esses classificadores [Tsoumakas et al. 2009a].

**Classifier Chain**. A abordagem CC se assemelha ao *BR* no sentido de envolver classificadores binários ao longo do seu aprendizado. Nela, são construídos classificadores binários ligados ao longo de uma cadeia, de maneira que os classificadores posteriores são induzidos a partir de um rótulo previsto por um classificador anterior na sequência. Com isso, para cada rótulo é criado um classificador binário, como na abordagem *BR*, e a cada predição, será levado em conta não só os termos, mas também os rótulos preditos pelos classificadores anteriores, o que faz com que, seja levada em conta a correlação entre eles. Na Figura 3 é ilustrada a estrutura do *Classifier Chain*

Formalmente falando, O conjunto de dados é transformado em $|L|$ conjuntos de dados, em que as instâncias do $j$-ésimo conjunto de dados têm o formato $((\mathbf{d}_i, r_1, ..., r_{j-1}), r_j)$, $r_j \in \{0, 1\}$. Assim, os classificadores constroem uma cadeia onde cada um deles aprende a classificação binária de um único rótulo, porém, considerando outros rótulos como atributos. Ao classificar novas instâncias, os rótulos são novamente previstos através da construção de uma cadeia de classificadores. A classificação começa com o primeiro classificador $C_1$ e prossegue para o último $C_{|L|}$, passando informações de rótulo entre os classificadores através do espaço de destaque. Portanto, a dependência entre rótulos é

preservada. No entanto, o resultado pode variar para diferentes ordens de cadeias [Rokach 2010].

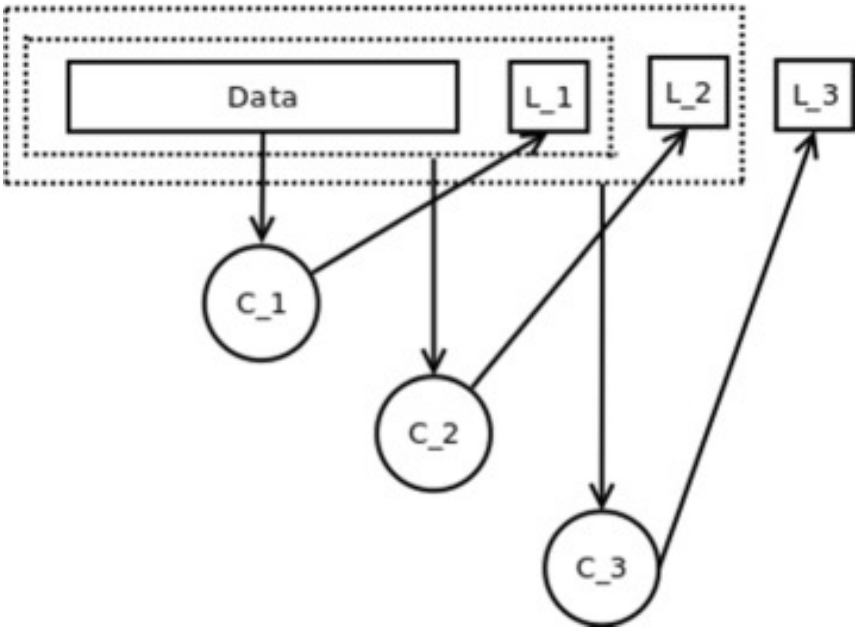



**Figura 3. Exemplo da estrutura do Classifier Chain [Riemenschneider et al. 2017]**

**Multi-Label Stacking**. Por fim, a abordagem MLS, esta é também conhecida como $2BR$, por utilizar uma duplicação do *Binary Relevance*. Com isso, é treinado para cada rótulo um classificador de base binária. Em seguida, o resultado (ou seja, as pontuações de confiança) desses classificadores é inserido em uma máquina de vetores de suporte (SVM) para melhorar a precisão da previsão. Esta abordagem é uma forma de considerar a correlação dos rótulos, algo que é criticado por não acontecer no $BR$. [Moyano et al. 2018],[Grigorios Tsoumakas 2009].

### 2.3.2. Algoritmos Próprios

Os algoritmos próprios considerados mais relevantes para este trabalho foram: *Multi-Label k-Nearest Neighbors* (ML*k*-NN) [Zhang and Zhou 2007], *Hierarchy Of Multi-label Classifiers* (HOMER) [Tsoumakas and Katakis 2007] e *BackPropagation for Multi-Label Learning* (BPMLL) [Zhang and Zhou 2006]. Abaixo serão apresentados cada um destes algoritmos citados.

**Multi-Label *k* Nearest Neighbors**. No MLk-NN são obtidos os $k$ vizinhos mais próximos de um documento a ser classificado. Após isso, é obtido um vetor de contagem, o qual armazena quantas vezes cada um dos rótulos da coleção apareceu no conjunto dos $k$ vizinhos mais próximos. Esse vetor de contagem é utilizado como um vetor de características que será submetido ao algoritmo *Naive Bayes* ($NB$). A partir daí, para cada rótulo $l \in L$, é executado o NB para definir se será atribuído o valor 0 ou 1 ao rótulo $l$.

**Hierarchy of Multi Label ClassifiER**. O HOMER é um algoritmo projetado para problemas grandes. Este constrói uma hierarquia de classificadores multirrótulo, de

maneira que cada um se torna responsável por um subconjunto disjunto de rótulos. Isso diminui sua complexidade computacional. Cada nó folha conterá uma única classe e os nós superiores aos nós folhas contém um classificador multirrótulo para um determinado conjunto de rótulos. Além disso, é realizado um balanceamento no número de exemplos e seus rótulos para cada classificador de forma a aumentar a performance de classificação [Tsoumakas et al. 2008]. Na Figura 4 é apresentada um exemplo de uma hierarquia de rótulos e respectivos classificadores construída pelo $HOMER$. Nesta figura, cada nó da hierarquia é um classificador multirrótulo capaz de predizer um ou mais rótulos (nós filhos).

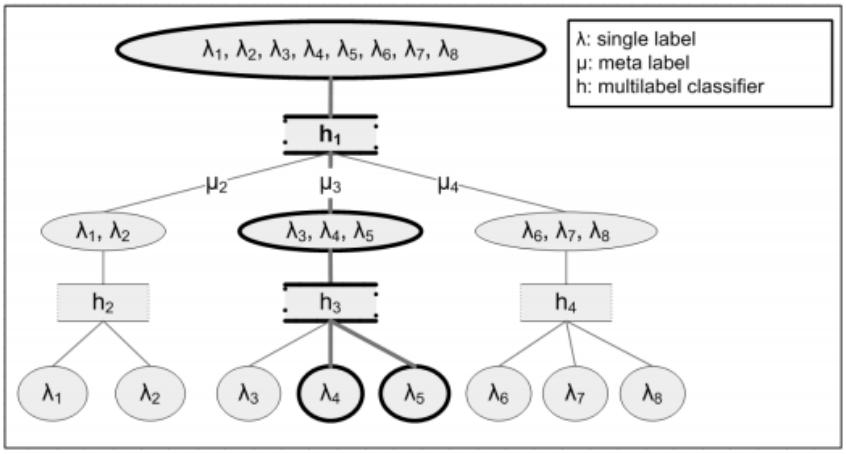



**Figura 4. Exemplo de uma hierarquia de classificadores. Fonte: [Tsoumakas et al. 2008].**

**BackPropagation for Multilabel Learning**. O algoritmo **BP-MLL** é derivado do popular algoritmo de retro-propagação (*Back Propagation* – BP) para redes neurais. Cada neurônio da camada de saída corresponderá à um rótulo. O valor de saída do neurônio é utilizando como valor de ranking para a classe. A partir daí, é utilizado um limiar de forma que se um neurônio de um rótulo disparar um valor acima deste limiar, o exemplo será classificado como pertence a classe. Além disso, uma das inovações do algoritmo **BP-MLL** é que esse limiar é inferido automaticamente por meio de um algoritmo de regressão linear [Zhang and Zhou 2006]. Na Figura 5 é ilustrada a estrutura da rede neural do BP-MLL.

## 3. Método de Pesquisa

Para atingir os objetivos deste trabalho, o método de pesquisa foi divido em 3 passos: (i) coleta de coleções de texto multirrótulo; (ii) aplicação de algoritmos de aprendizado multirrótulo; e (iii) avaliação da performance de classificação. Todos os passos do método utilizado serão detalhados nas próximas subseções. Vale ressaltar que foi desenvolvido um produto de *software* (no formato de um *framework*) que implementa os passos `ii` e `iii`.

### 3.1. Coleta de Coleções de Textos Multirrótulo

Como etapa inicial do método de pesquisa, e para viabilizar as etapas subsequentes, foram coletadas bases textuais multirrótulo, sendo elas: Bibtex

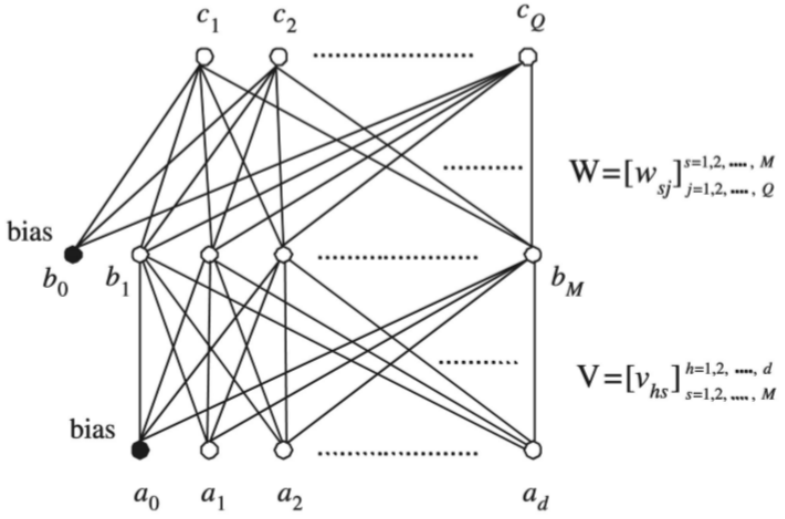



**Figura 5. Exemplo da arquitetura da rede neural do BP-MLL. Fonte: [Zhang and Zhou 2006].**

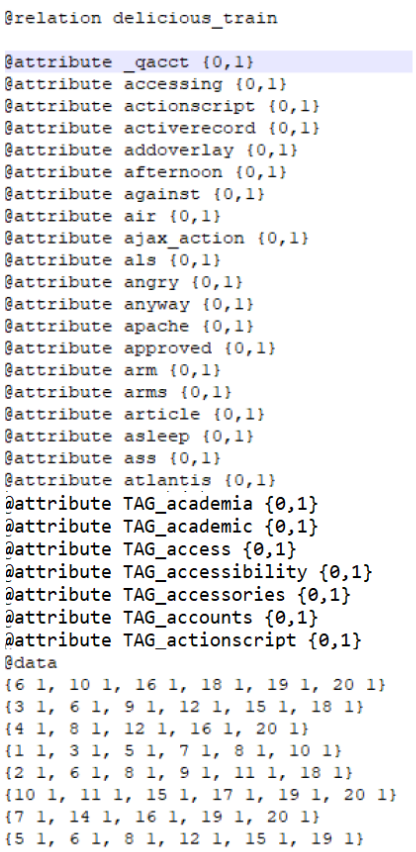
[Katakis et al. 2008], Delicious [Tsoumakas et al. 2008], Enron [Berkeley ], Medical [Pestian et al. 2007], Rcv1v2subset1, Rcv1v2subset2, Rcv1v2subset3, Rcv1v2subset4, Rcv1v2subset5 [Lewis et al. 2004], Tmc2007 [Srivastava and Zane-Ulman 2005] e Yahoo [Ueda and Saito 2003]. Todas as bases coletadas já estavam pré processadas e nos formatos ARFF e XML, conforme apresentado nas Figuras 6 e 7 respectivamente.

As representações *bag-of-words* das coleções coletadas estão no formato ARFF, mesmo formato da biblioteca Mulan, a qual é utilizada em outras etapas do *framework* proposto [Tsoumakas and Katakis 2007]. Na Figura 6 é apresentado um exemplo da estrutura do formato ARFF. A primeira linha (`@relation`) contém o nome da base de dados. As linhas subsequentes contém descritores dos atributos da representação (`@attribute`). Vale ressaltar que muitos dos algoritmos da ferramenta Mulan consideram apenas atributos binários, isto, é, indicam apenas a ocorrência ou ausência de um termo no atributo. Caso o algoritmo esteja apto a considerar atributos com pesos reais, no lugar de "0,1" deve se colocar a palavra "NUMERIC". Após listados os atributos, têm-se as informações sobre os pesos dos termos nos documentos da coleção, que serão apresentados na área de dados (`@data`). No exemplo apresentado na Figura 6, a representação está no formato esparso, o qual apresenta apenas os índices dos atributos e seus respectivos pesos para aqueles cujo peso é diferente de 0.

Além do formato ARFF para representar a coleção de texto, é também necessário um arquivo auxiliar para indicar quais dos atributos da representação são rótulos de categorias e consequentemente serão utilizados como atributos alvos no momento do aprendizado. Para isso, é gerado um arquivo *XML* (*eXtensible Markup Language*), o qual armazena quais atributos da representação correspondem às classes dos documentos. Um exemplo desse arquivo no formato XML é apresentado na Figura 7.

### 3.2. Aplicação de Algoritmos de Aprendizado Multirrótulo

Nesta subseção é exibida a configuração experimental, mostrando as bases utilizadas na realização dos testes, os algoritmos de transformação do problema e algoritmos de apren-



```
@relation delicious_train

@attribute _qacct {0,1}
@attribute accessing {0,1}
@attribute actionscript {0,1}
@attribute activerecord {0,1}
@attribute addoverlay {0,1}
@attribute afternoon {0,1}
@attribute against {0,1}
@attribute air {0,1}
@attribute ajax_action {0,1}
@attribute als {0,1}
@attribute angry {0,1}
@attribute anyway {0,1}
@attribute apache {0,1}
@attribute approved {0,1}
@attribute arm {0,1}
@attribute arms {0,1}
@attribute article {0,1}
@attribute asleep {0,1}
@attribute ass {0,1}
@attribute atlantis {0,1}
@attribute TAG_academia {0,1}
@attribute TAG_academic {0,1}
@attribute TAG_access {0,1}
@attribute TAG_accessibility {0,1}
@attribute TAG_accessories {0,1}
@attribute TAG_accounts {0,1}
@attribute TAG_actionscript {0,1}
@data
{6 1, 10 1, 16 1, 18 1, 19 1, 20 1}
{3 1, 6 1, 9 1, 12 1, 15 1, 18 1}
{4 1, 8 1, 12 1, 16 1, 20 1}
{1 1, 3 1, 5 1, 7 1, 8 1, 10 1}
{2 1, 6 1, 8 1, 9 1, 11 1, 18 1}
{10 1, 11 1, 15 1, 17 1, 19 1, 20 1}
{7 1, 14 1, 16 1, 19 1, 20 1}
{5 1, 6 1, 8 1, 12 1, 15 1, 19 1}
```

**Figura 6. Exemplo de documento ARFF da base delicious (resumida)**

dizado multi-classe usados em conjunto com os métodos de adaptação do problema, e os algoritmos próprios. Também são apresentados os parâmetros utilizados para cada algoritmo/abordagem, o esquema e as métricas de avaliação.

1. **Coleções de textos:**
   **Bibtex**.
   **Business1**.
   **Society1**.
2. **Algoritmos de Transformação do problema:**
   **Label Power Set(LP)**.
   **Binary Relevance (BR)**.
   **Classifier Chain (CC)**.
   **Ensembles of Pruned Sets (EPS)**.
   **Multi-Label Stacking (MLS)**.
3. **Algoritmos de aprendizado multi-classe utilizados em conjunto com os métodos de adaptação do problema:**
   **k-Nearest Neighbors (k-NN)** [Peterson 2009]. Foi considerado $k$ = 1,3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 25, 29, 35, 41, 49, 57, 73, 89 com e sem voto ponderado pela distância, e cosseno como medida de proximidade.
   **J48** [Patil et al. 2013]. Níveis de confiança para poda da árvore = 0.15, 0.20, 0.25.

```xml
<?xml version="1.0" encoding="utf-8"?>
<labels xmlns="http://mulan.sourceforge.net/labels">
<label name="TAG_.imported"></label>
<label name="TAG_.net"></label>
<label name="TAG_2.0"></label>
<label name="TAG_2007"></label>
<label name="TAG_3d"></label>
<label name="TAG_??"></label>
<label name="TAG_???"></label>
<label name="TAG_????"></label>
<label name="TAG_academia"></label>
<label name="TAG_academic"></label>
<label name="TAG_access"></label>
<label name="TAG_accessibility"></label>
<label name="TAG_accessories"></label>
<label name="TAG_accounts"></label>
<label name="TAG_actionscript"></label>
<label name="TAG_activism"></label>
<label name="TAG_ad"></label>
<label name="TAG_addon"></label>
<label name="TAG_addons"></label>
<label name="TAG_admin"></label>
<label name="TAG_administration"></label>

</labels>
```

**Figura 7. Exemplo de documento XML o qual define quais atributos do ARFF correspondem aos rótulos da base.**

**Sequential Minimal Optimization (SMO)** [Platt 1998]. Para este, foi considerado CS = 0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000, 100000 *Kernel* = Linear, Polynomial, RBF.

**Multinomial Naïve Bayes (MNB)** [Kibriya et al. 2004]. Foram utilizados os valores padrões da biblioteca Weka [John and Langley 1995]

**Inductive Model based on Heterogeneous Network (IMBHN)** [Rossi et al. 2016]. Taxa de correção de erro = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, número máximo de iterações = 1000, e erro quadrático médio mínimo = 0.01, 0.005.

4. **Algoritmos próprios:**

   **Multilabel k-Nearest Neighbors (Ml*k*-NN)**. Foi considerado $k$ = 1,3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 25, 29, 35, 41, 49, 57, 73, 89 com e sem voto ponderado pela proximidade, e cosseno como medida de proximidade.

   **Hierarchy of Multilabel Classifier (HOMER)**. Foram utilizados os valores padrões da biblioteca Mulan.

   **BackPropagation for Multi-Label Learning (B-PMLL)**. Para este foram utilizados os valores padrão da biblioteca Mulan, sendo eles: taxa de aprendizado = 0.05, regularização da deterioração de pesos = 0.00001 e período de treinamento = 100, e normalização dos valores dos atributos.

### 3.3. Avaliação da Performance de Classificação

Existem diversas medidas para a classificação multirrótulo. Estas se dividem em dois grandes grupos: (i) baseadas em exemplo e (ii) baseadas em rótulo. As medidas baseadas em exemplo, calculam a performance do classificador para cada exemplo separadamente, para depois calcular uma média dos exemplos. Já as medidas baseadas em rótulo são calculadas fazendo uso de medidas de avaliação binárias monorrótulo, que são aplicadas para

cada um dos rótulos separadamente, após isso uma média de todas estas medidas binárias é calculada. Na Figura 8 são apresentadas as medidas de avaliação que são utilizadas no *framework* proposto. Vale ressaltar que todas estas medidas vem sendo utilizadas na literatura.

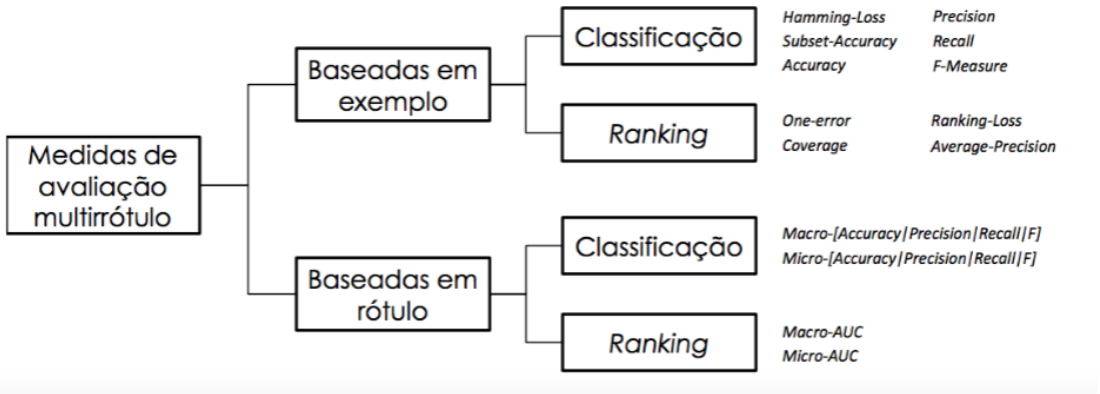



**Figura 8. Medidas de avaliação [Cherman 2013].**

Neste trabalho foram consideradas as medidas Macro-F1 e *Hamming Loss*, i.e., uma de cada categoria. A medida Macro-F1 considera o acerto exato do rótulo e cada possível combinação de rótulo para realizar uma média da precisão e revocação de tal rótulo. Nessa medida, quanto maior o seu valor, maior o acerto do classificador. Já a medida *Hamming Loss* mede a fração de rótulos errados por utilizando a função `xor` para cada rótulo real e cada rótulo predito. Nessa medida, quanto menor o valor, maior o acerto do classificador. Os valores de ambas medida apresentadas nessa seção correspondem à uma média dos valores obtidos no procedimento de validação cruzada em 10 pastas [Tan et al. 2006].

## 4. Resultados e Discussão

Nesta seção serão apresentadas informações e resultados da avaliação experimental realizada neste projeto. Serão apresentadas 3 seções: resultados da medida Macro-F1, resultados da medida Hamming-Loss, e por fim, as discussões dos resultados.

### 4.1. Resultados da Medida Macro-F1

Nesta Seção são apresentados os resultados obtidos considerando a medida Macro-F1. Na Figura 9 e Tabela 4 são apresentados os resultados para a base Bibtex considerando algoritmos de transformação do problema. De acordo com esta figura, pode-se notar que os melhores resultados foram obtidos por algoritmos de transformação *Classifier Chain* e *Binary Relevance* juntamente com o algoritmo *k-NN*.

Já na Figura 10 e Tabela 5 são apresentados os resultados para a base Bibtex considerando algoritmos de aprendizado multirrótulo próprios. Como é possível observar, os algoritmos ML*k*-NN e HOMER obtiveram os melhores resultados, obtendo uma performance de classificação muito maior se comparado ao BPMLL.

Ao comparar os resultados dos algoritmos de transformação do problema com os algoritmos próprios, pode-se observar que os algoritmos de transformação obtiveram per-

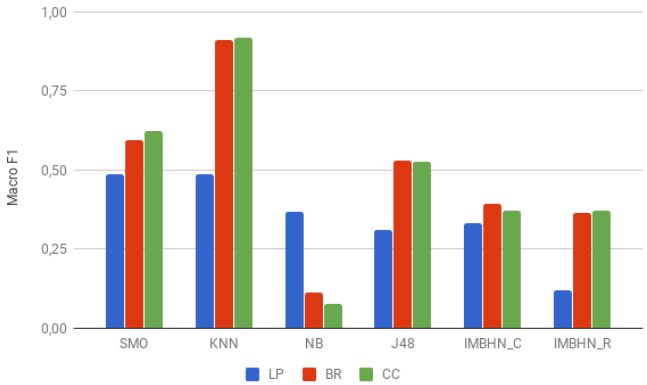



**Figura 9. Algoritmos de transformação do problema na base Bibtex.**

**Tabela 4. Algoritmos de adaptação para a base Bibtex, com a medida Macro-F1**

| | LP | BR | CC |
|---|---|---|---|
| **SMO** | 0,4851 | 0,5936 | 0,6226 |
| **KNN** | 0,4851 | 0,9113 | 0,9162 |
| **MNB** | 0,3666 | 0,1123 | 0,0761 |
| **J48** | 0,3097 | 0,5276 | 0,5264 |
| **IMBHN_C** | 0,3300 | 0,3920 | 0,3700 |
| **IMBHN_R** | 0,1204 | 0,3627 | 0,3704 |

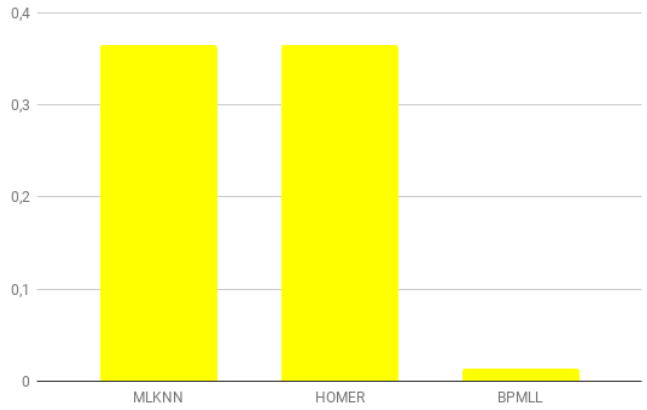



**Figura 10. Algoritmos próprios na base Bibtex.**

formances de classificação superiores, conforme ilustrado na Figura 11, a qual apresenta um comparativo entre o Homer e o BR com *k*-NN.

Na Figura 12 e Tabela 7 são apresentados os resultados da base Business1 para os algoritmos de transformação. Para esta base, pode-se observar que o IMBHN unido com BR e CC obtiveram os melhores resultados entre os algoritmos próprios. Já na Figura 13 são apresentados os resultados dos algoritmos próprios. É possível verificar que entre os algoritmos próprios, ML*k*-NN e HOMER apresentaram os melhores resultado para a base Business1.

**Tabela 5. Algoritmos próprios para a base Bibtex, com a medida Macro-F1**

| | **Macro-F1** |
|---|---|
| **ML*k*-NN** | 0,3646 |
| **HOMER** | 0,3646 |
| **BP-MLL** | 0,0136 |

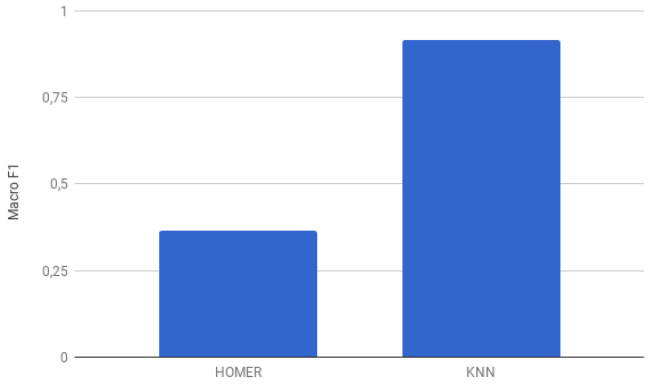



**Figura 11. Comparativo do melhor algoritmo de transformação e melhor algoritmo próprio na base Bibtex.**

Na Figura 14 e Tabela 8 são apresentados os resultados para a base Society1. Pode-se observar para esta base um destaque para a medida MLS em conjunto com o algoritmo IMBHN_R, conjunto o qual obteve os melhores resultados. Já a abordagem LP objeteve a pior performance em conjunto com o mesmo algoritmo.

Na Figura 15 e Tabela 9 são apresentados os resultados dos algoritmo próprios para a base Society1. É possível observar os algoritmo MLk-NN apresentou resultados muito superiores ao algoritmo HOMER para esta base.

## 4.2. Resultados da Medida *Hamming Loss*

Nesta Seção são apresentados os resultados obtidos considerando a medida *Hamming Loss*. Na Figura 16 e Tabela 10 são apresentados os resultados dos algoritmos de

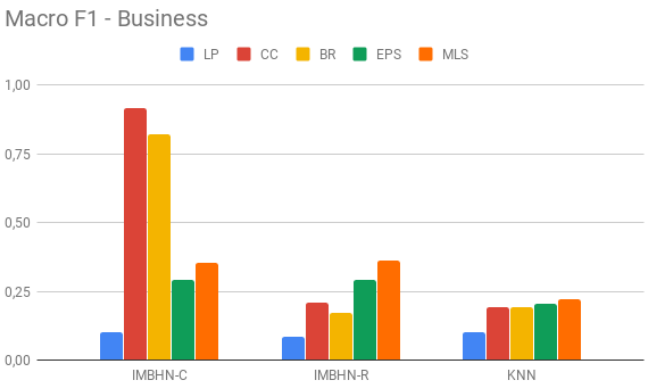



**Figura 12. Comparativo entre algoritmos de transformação para a base Business1**

**Tabela 6. Algoritmos de adaptação para a base Business1, com a medida Macro-F1**

| | LP | BR | CC | EPS | MLS |
|---|---|---|---|---|---|
| **IMBHN_C** | 0,1030 | 0,9160 | 0,8220 | 0,2930 | 0,3540 |
| **IMBHN_R** | 0,0840 | 0,2100 | 0,1710 | 0,2930 | 0,3610 |
| **KNN** | 0,1000 | 0,1922 | 0,1928 | 0,2063 | 0,2198 |

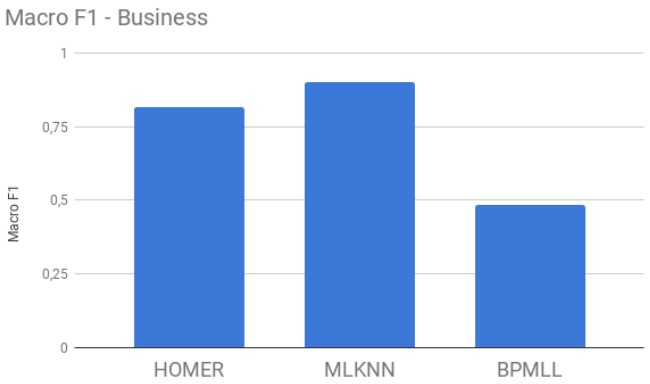



**Figura 13. Comparativo entre algoritmos próprios para a base Business1**

adaptação de problema. Observa-se que a maioria das abordagens tiveram resultados próximos para diferentes classificadores. Porém, novamente, os melhores resultados foram para as abordagens BR e CC.

Na Figura 19 e Tabela 13 são apresentados os resultados da base Bibtex com os algoritmos próprios. Nessa avaliação, pode-se observar que o *MLk-NN* obteve os melhores resultados.

Na Figura 18 e Tabela 12 são apresentados os resultados dos algoritmos de adaptação de problema para a base Business1. Destaca-se nessa avaliação a abordagem EPS apresentou melhores resultados em relação a outras abordagens.

Na Figura 19 e Tabela 13 são apresentados os resultados da base Business1 com

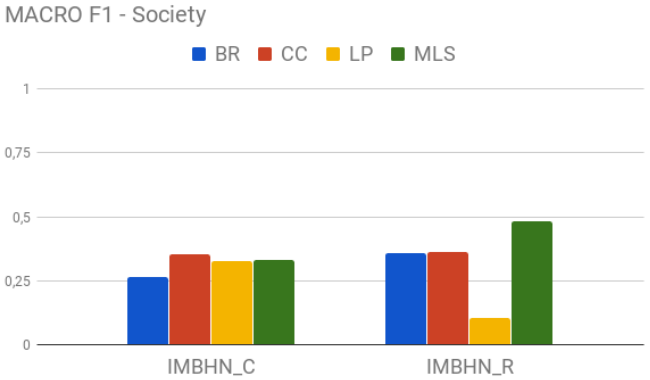



**Figura 14. Macro-F1 resultante de algoritmos de adaptação para base Society1**

**Tabela 7. Algoritmos próprios para a base Business1 com a medida Macro-F1**

| | Macro-F1 |
|---|---|
| **ML*k*-NN** | 0,8160 |
| **HOMER** | 0,9010 |
| **BP-MLL** | 0,4830 |

**Tabela 8. Algoritmos de adaptação para a base Society1, com a medida Macro-F1**

| | LP | BR | CC | MLS |
|---|---|---|---|---|
| **IMBHN_C** | 0,265 | 0,3534 | 0,3253 | 0,3323 |
| **IMBHN_R** | 0,3582 | 0,3613 | 0,1053 | 0,4825 |

os algoritmos próprios. Nessa avaliação, pode-se observar que o HOMMER obteve os melhores resultados.

Na Figura 20 e Tabela 14 são apresentados os resultados dos algoritmos de adaptação de problema para a base Society1. Nesta avaliação, a abordagem *CC* obteve os melhores resultados.

Por fim, na Figura 21 e Tabela 15 são apresentados os resultados dos algoritmos de adaptação de problema para a base Society1. Observa-se que o HOMER obteve os melhores resultados.

### 4.3. Discussões

Apesar de não ter gerado os resultados considerando todos os algoritmos e todas as bases, pode-se fazer algumas observações acerca dos resultados. Considerando os algoritmos de adaptação do problema, pôde-se observar que:

- A abordagem LP proveu resultados inferiores em relação às outras abordagens, portanto, não sendo aconselhável o seu uso na prática.
- A abordagens CC e BR apresentaram resultados competitivos em relação à outras abordagens. Porém, como não houve grandes diferenças de performance entre ambas, o BR seria aconselhável na prática já que possui um custo menor que o CC, além de ser mais fácil de ser implementado.

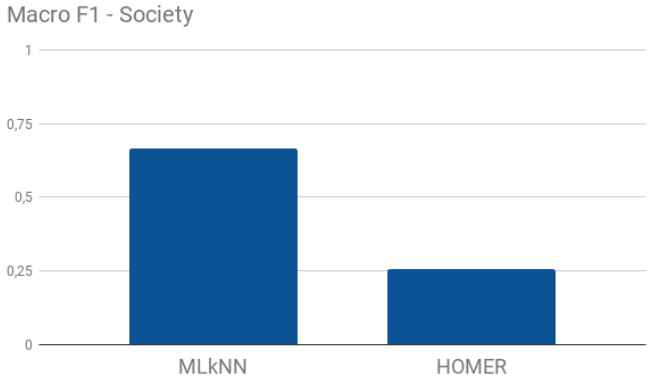



**Figura 15. Macro-F1 resultante de algoritmos próprios para a base Society1**

**Tabela 9. Algoritmos próprios para a base Society1, com a medidas Macro-F1**

| | Macro-F1 |
|---|---|
| **ML*k*-NN** | 0,6670 |
| **HOMER** | 0,2560 |

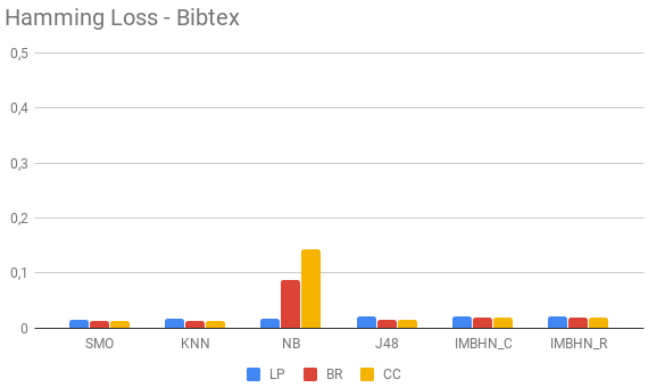



**Figura 16. Gráfico do *Hamming Loss* algoritmos de transformação do problema na base Bibtex.**

- A abordagem MLS teve apresentou alguns resultados que se destacaram para algumas bases. Como sua complexidade não é tão maior que a complexidade do BR, também é uma abordagem interessante na prática, uma vez que esse tipo de abordagem é capaz de modelar o relacionamento entre os rótulos.

  Já considerando os algoritmos próprios, pôde-se observar que:

- A abordagem baseada em redes neurais apresentou resultados inferiores em relação as demais abordagens.
- No, o algoritmo ML$k$NN e o HOMER apresentaram as melhores performances. Entretanto, vale ressaltar que o algoritmo ML$k$NN é menos custoso que o algoritmo HOMER. Porém, dependendo da necessidade da aplicação, por exemplo, a geração de respostas de classificação rápidas, o HOMER seria mais indicado que o ML$k$NN.

No geral, comparando as duas abordagens (algoritmos próprios e adaptação do problema) e utilizando, na maioria das vezes os algoritmos de transformação do problema apresentaram performances superiores aos algoritmos próprios.

**Tabela 10. Resultados do *Hamming Loss* dos algoritmos de adaptação para a base Bibtex**

| | LP | BR | CC |
|---|---|---|---|
| **SMO** | 0,0154 | 0,0135 | 0,0132 |
| **KNN** | 0,0160 | 0,0132 | 0,0131 |
| **MNB** | 0,0172 | 0,0874 | 0,1419 |
| **J48** | 0,0206 | 0,0146 | 0,0146 |
| **IMBHN_C** | 0,0199 | 0,0182 | 0,0191 |
| **IMBHN_R** | 0,0201 | 0,0194 | 0,0190 |

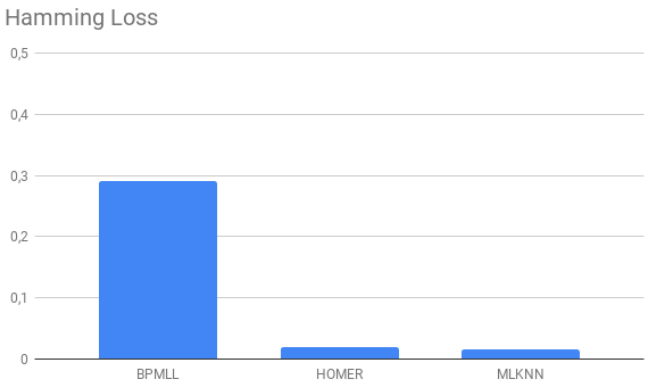



**Figura 17. Comparativo entre algoritmos próprios para a base Bibtex considerando a medida *Hamming Loss*.**

**Tabela 11. Resultados dos algoritmos próprios para a base Bibtex considerando a medida *Hamming Loss*.**

| | Hamming Loss |
|---|---|
| **ML*k*-NN** | 0,0151 |
| **HOMER** | 0,0188 |
| **BP-MLL** | 0,2908 |

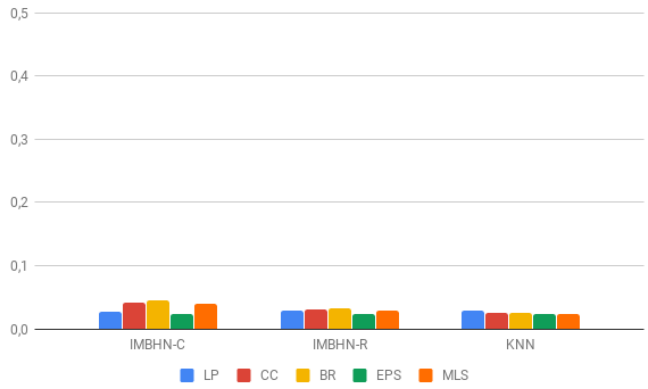



**Figura 18. Comparativo entre algoritmos de transformação para a base Business1 considerando a medida *Hamming Loss*.**

**Tabela 12. Resultados dos algoritmos de adaptação para a base Business1 considerando a medida *Hamming Loss*.**

| | LP | BR | CC | EPS | MLS |
|---|---|---|---|---|---|
| **IMBHN_C** | 0,0269 | 0,0419 | 0,0453 | 0,0232 | 0,0398 |
| **IMBHN_R** | 0,0281 | 0,0312 | 0,0321 | 0,0232 | 0,0293 |
| **KNN** | 0,0281 | 0,0248 | 0,0247 | 0,0232 | 0,0233 |

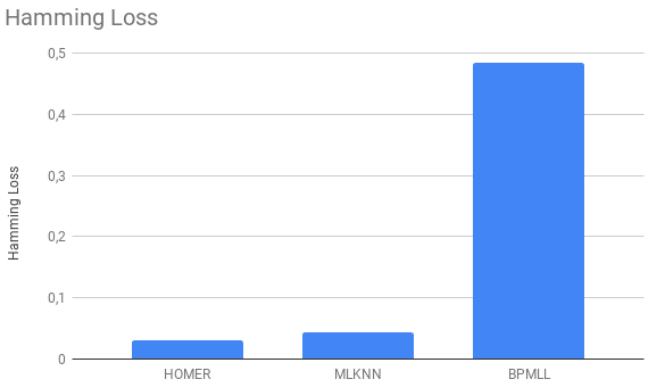



**Figura 19. Comparativo entre algoritmos próprios para a base Business1**

**Tabela 13. Algoritmos próprios para a base Business1**

| | Hamming Loss |
|---|---|
| **ML*k*-NN** | 0,0433 |
| **HOMER** | 0,0300 |
| **BP-MLL** | 0,4833 |

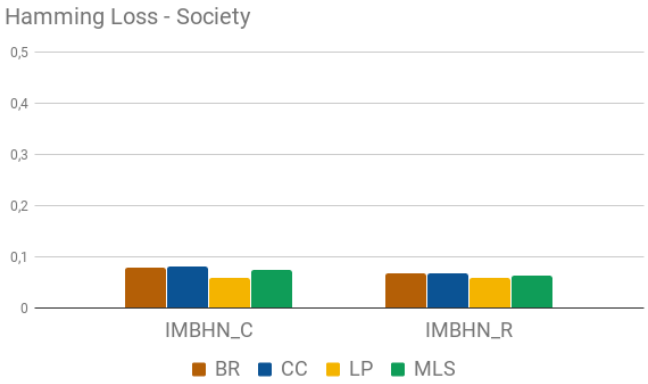



**Figura 20. Comparativo entre algoritmos de transformação para a base Society1**

**Tabela 14. Algoritmos de adaptação para a base Society1, com a medida Hamming Loss**

| | BR | CC | LP | MLS |
|---|---|---|---|---|
| **IMBHN_C** | 0,0774 | 0,0800 | 0,0581 | 0,0727 |
| **IMBHN_R** | 0,0679 | 0,067 | 0,0579 | 0,0629 |

**Tabela 15. Algoritmos próprios para a base Society1, com a medida Hamming Loss**

| | Hamming Loss |
|---|---|
| **ML*k*-NN** | 0,0781 |
| **HOMER** | 0,0656 |

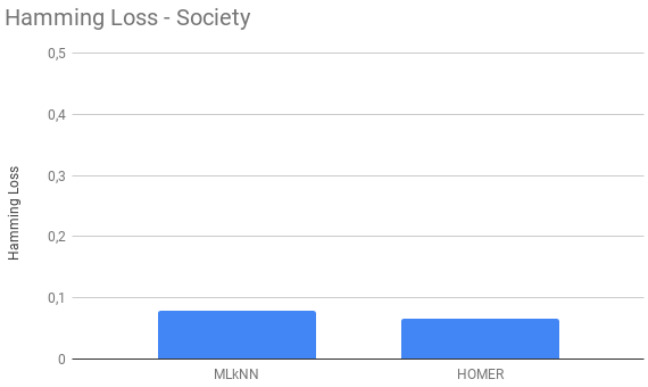



**Figura 21. Hamming Loss resultante de algoritmos próprios para a base Society1**

## 5. Conclusões

Com os resultados obtidos, observou-se que no geral, os métodos de adaptação de problema podem ter performance superior aos algoritmos próprios para aprendizado multirrótulo. Destacaram-se na avaliação realizada neste artigo os métodos métodos de adaptação *Binary Relevance*, *Classifier Chain* e *MultiLabel Stacking*, e os algoritmos próprios *Multi-Label k-Nearest Neighbors* e *Hierarchy of Multilabel Classifier*.

Apesar do *Classifier Chain* em alguns momentos apresentar melhores resultados com relação ao *Binary Relevance*, ele é muito mais custoso computacionalmente, o que faz com que o *Binary Relevance* seja o melhor custo-benefício em termos de complexidade computacional e performance de classificação. Vale ressaltar também que o *MultiLabel Stacking* não apresenta custo significativamente superior ao *Binary Relevance* e apresentou resultados competitivos com o mesmo, e, em algumas situações, muito superior. Porém, vale ressaltar que em diversos casos, os algoritmos próprios possuem desempenho igual ou similar aos algoritmos de transformação do problema, com menção principal ao *Multi-Label k-Nearest Neighbors* que apresentou os melhores e ao *Hierarchy of Multilabel Classifier*.

Como trabalhos futuros pretende-se finalizar a execução dos experimentos para que se possa chegar a análise mais conclusivas. Pretende-se também avaliar o uso de outras arquitetura de redes neurais no aprendizado multirrótulo para a classificação de textos.

## Referências


Alpaydin, E. (2009). *Introduction to machine learning*. MIT press.

Berkeley, U. Uc berkeley enron email analysis.

Cherman, E. A. (2013). *Aprendizado de máquina multirrótulo: explorando a dependência de rótulos e o aprendizado ativo*. PhD thesis, Universidade de São Paulo.

Dumais, S., Platt, J., Heckerman, D., and Sahami, M. (1998). Inductive learning algorithms and representations for text categorization.

Grigorios Tsoumakas, Anastasios Dimou, E. S. V. M. I. K. I. V. (2009). Correlation-based pruning of stacked binary relevance models for multi-label learning. In *Proc. ECML/PKDD 2009 Workshop on Learning from Multi-Label Data (MLD'09)*, pages 101–116.

John, G. H. and Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. In *Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345, San Mateo. Morgan Kaufmann.

Katakis, I., Tsoumakas, G., and Vlahavas, I. (2008). Multilabel text classification for automated tag suggestion. In *Proceedings of the ECML/PKDD*, volume 18, page 5.

Kibriya, A. M., Frank, E., Pfahringer, B., and Holmes, G. (2004). Multinomial naive bayes for text categorization revisited. In *Australasian Joint Conference on Artificial Intelligence*, pages 488–499. Springer.

Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397.

Luaces, O., Díez, J., Barranquero, J., del Coz, J. J., and Bahamonde, A. (2012). Binary relevance efficacy for multilabel classification. *Progress in Artificial Intelligence*, 1(4):303–313.

Manning, C., Raghavan, P., and Schütze, H. (2010). Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103.

Matsubara, E. T., Martins, C. A., and Monard, M. C. (2003). Pretext: Uma ferramenta para pré-processamento de textos utilizando a abordagem bag-of-words. *Techinical Report*, 209:4.

Moyano, J. M., Gibaja, E. L., Cios, K. J., and Ventura, S. (2018). Review of ensembles of multi-label classifiers: models, experimental study and prospects. *Information Fusion*, 44:33–45.

Patil, T. R., Sherekar, S., et al. (2013). Performance analysis of naive bayes and j48 classification algorithm for data classification. *International journal of computer science and applications*, 6(2):256–261.

Pestian, J. P., Brew, C., Matykiewicz, P., Hovermale, D. J., Johnson, N., Cohen, K. B., and Duch, W. (2007). A shared task involving multi-label classification of clinical free text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 97–104. Association for Computational Linguistics.

Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia*, 4(2):1883.

Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines.

Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2011). Classifier chains for multi-label classification. *Machine learning*, 85(3):333.

Riemenschneider, M., Herbst, A., Rasch, A., Gorlatch, S., and Heider, D. (2017). ecccl: parallelized gpu implementation of ensemble classifier chains. *BMC bioinformatics*, 18(1):371.

Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39.

Rossi, R. G., de Andrade Lopes, A., and Rezende, S. O. (2016). Optimization and label propagation in bipartite heterogeneous networks to improve transductive classification of texts. *Information Processing & Management*, 52(2):217–257.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Sorower, M. S. (2010). A literature survey on algorithms for multi-label learning. *Oregon State University, Corvallis*, 18:1–25.

Srivastava, A. N. and Zane-Ulman, B. (2005). Discovering recurring anomalies in text reports regarding complex space systems. In *2005 IEEE aerospace conference*, pages 3853–3862. IEEE.

Tan, P.-N., Steinbach, M., and Kumar, V. (2006). Introduction to data mining, pearson education. *Inc., New Delhi*.

Tsoumakas, G., Dimou, A., Spyromitros, E., Mezaris, V., Kompatsiaris, I., and Vlahavas, I. (2009a). Correlation-based pruning of stacked binary relevance models for multi-label learning. In *Proceedings of the 1st international workshop on learning from multi-label data*, pages 101–116.

Tsoumakas, G. and Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13.

Tsoumakas, G., Katakis, I., and Vlahavas, I. (2008). Effective and efficient multilabel classification in domains with large number of labels. In *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, volume 21, pages 53–59. sn.

Tsoumakas, G., Katakis, I., and Vlahavas, I. (2009b). Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer.

Turner, V., Gantz, J. F., Reinsel, D., and Minton, S. (2014). The digital universe of opportunities: Rich data and the increasing value of the internet of things. *IDC Analyze the Future*, 16.

Ueda, N. and Saito, K. (2003). Parametric mixture models for multi-labeled text. In *Advances in neural information processing systems*, pages 737–744.

Wallach, H. M. (2006). Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984. ACM.

Weiss, S. M., Indurkhya, N., and Zhang, T. (2015). *Fundamentals of predictive text mining*. Springer.

Zhang, M.-L. and Zhou, Z.-H. (2006). Multilabel neural networks with applications to functional genomics and text categorization. *IEEE transactions on Knowledge and Data Engineering*, 18(10):1338–1351.

Zhang, M.-L. and Zhou, Z.-H. (2007). Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048.

Zhang, M.-L. and Zhou, Z.-H. (2013). A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837.